

# Graph Neural Networks with Multiscale Soft Attention

Studierende: B. Sc.  
Verantwortlicher Hochschullehrer: Prof. Dr.-Ing. Patrick Mäder (JP)  
Betreuender Assistent: Dr. rer.nat. Marco Seeland  
Fachgebiet: Softwaretechnik für sicherheitskritische Systeme  
Dauer: (von) - (bis)

## Beschreibung

In der Einbeziehung nicht-euklidischer Strukturinformationen für Aufgaben des Maschinellen Lernens stellen Neuronale Graph Netzwerke den aktuellen Stand der Technik dar. Die erreichten Ergebnisse mit aufmerksamkeitsbasierten Graph Netzwerken (engl. *Graph Attention Networks*) [1] zeigen zusätzlich, dass Knotenmerkmale bzw. -repräsentationen nicht gleichmäßig sondern gewichtet innerhalb einer Nachbarschaft ausgetauscht werden sollten. Für die Berechnung aufmerksamkeitsbasierter Knotenrepräsentationen werden üblicherweise nur die nächsten Nachbarn eines Knotens innerhalb eines Graphen betrachtet, d.h. nur Nachbarn erster Ordnung. Gleichzeitig hat sich für faltende Graph Netzwerke (engl. *Graph Convolutional Networks*) gezeigt, dass die unmittelbare Einbeziehung von Nachbarschaften höherer Ordnungen vorteilhaft ist und effizienter gegenüber tiefen Netzen [2, 3].

Die Zielstellung dieser Arbeit liegt in der Erweiterung der Aufmerksamkeitsmechanismen in Graph Netzwerken, sodass auch Nachbarn höherer Ordnungen einbezogen werden können. Algebraisch kann dies einfach durch Potenzierung der (ungewichteten) Adjazenzmatrix  $A$  erfolgen:

$$\hat{A}^{(K)} = A + \sum_{k=2}^K \frac{1}{k} \left( (A^k > 0) - A \right). \quad (1)$$

Die transformierte Adjazenzmatrix  $\hat{A}^{(K)}$  enthält dann alle Nachbarn bis zur  $K$ -ten Ordnung, gewichtet mit dem rekursiven Grad der jeweiligen Nachbarschaft, und kann anstelle der originalen Adjazenzmatrix verwendet werden. Alternativ können Nachbarschaften unterschiedlicher Ordnungen  $k = 2, \dots, K$

$$\hat{A}^{(k)} = (A^k > 0) - A \quad (2)$$

in  $K$  verschiedenen Netzwerkästen separat ausgewertet und anschließend durch geeignete Aggregationsmechanismen (Pooling oder Aneinanderreihung) zusammengeführt werden.

## Aufgabenstellung

Nach einer Literaturrecherche sollen aktuelle aufmerksamkeitsbasierte Graph Netzwerke um die Einbeziehung von Nachbarschaften höherer Ordnungen erweitert werden. Mindestens sollen dabei die auf Bahdanau Attention basierenden *Graph Attention Networks* (GAT) [1] und *Graph Transformer* [4] untersucht werden.

Die Erweiterungen und ihre jeweiligen Ausprägungsformen sollen systematisch durch Experimente evaluiert werden. Prototypisch soll dafür die Aufgabe der Knotenklassifikation auf gängigen Datensätzen (Cora, Citeseer, Pubmed, OGB, ...) ausgewertet werden. Durch Auswertung der Attentiongewichte soll untersucht werden, inwiefern Nachbarn höherer Ordnungen einbezogen werden, und ob eine separierte Berechnung durch  $K$  Äste Vor- bzw. Nachteile gegenüber einer einmaligen Berechnung mittels  $\hat{A}^{(K)}$  hat. Darüber hinaus ist zu untersuchen, wie der maximale Grad der Nachbarschaft mit der potentiellen Verringerung der Netzwerktiefe korreliert.

Das Projekt soll in Python unter Verwendung der Bibliothek PyTorch Geometric [5] entwickelt werden. Die Abgabe erfolgt durch Übergabe des dokumentierten Projekts und einer schriftlichen Ausarbeitung von maximal 15 Seiten.

**Schlagwörter:** Graph Neural Networks; Attention

Wochen	Aufgaben
1-3	Literaturrecherche, Einarbeitung in Stand der Technik und Methodenauswahl
...	...
20-24	Aufbereitung der Ergebnisse und Verfassen der Arbeit

## Literaturverzeichnis

- [1] P. Velicković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, “Graph attention networks,” oct 2017.
- [2] S. Abu-El-Haija, B. Perozzi, A. Kapoor, N. Alipourfard, K. Lerman, H. Harutyunyan, G. V. Steeg, and A. Galstyan, “MixHop: Higher-Order Graph Convolutional Architectures via Sparsified Neighborhood Mixing,” tech. rep., may 2019.
- [3] E. Rossi, F. Frasca, D. Eynard, B. Chamberlain, M. Bronstein, and F. Monti, “SIGN: Scalable inception graph neural networks,” *arXiv*, 2020.
- [4] V. P. Dwivedi and X. Bresson, “A generalization of transformer networks to graphs,” 2020.
- [5] M. Fey and J. E. Lenssen, “Fast graph representation learning with PyTorch Geometric,” in *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.